UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/721,300	11/24/2003	Kenway W. Tam	SUNMP351	7065

32291 7590 04/12/2007

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER	
LAI, VINCENT	
ART UNIT	**PAPER NUMBER**
2181	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/12/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/721,300	TAM ET AL.	
	Examiner	Art Unit	
	Vincent Lai	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 January 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,5-12 and 14-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,5-12 and 14-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 29 January 2007 has been entered.

Response to Amendment

2. Claim objections, 35 USC 101 rejections and 35 USC 112 rejections have been withdrawn after considering amendments.

Response to Arguments

3. Applicant's arguments filed 29 January 2007 have been fully considered but they are not persuasive.

Applicant argues, "Hennessey's MOVI2S, MOVS2I and any immediately subsequent move instruction will have a progressive latency that progressively gets worse as described in Applicant's paragraph 4-6 and Figure 1A-1B."

It is noted that a move instruction, such as MOVI2S and MOVS2I, is not a swap instruction, as Applicant appears to equate. It is also noted that such reasoning is a conjecture of the Applicant, as Hennessey does not disclose such drawbacks. The swap request of the Applicant is essentially two move instructions that utilize a special register. The MOVI2S and MOVS2I instructions work in a very similar fashion and accordingly does not appear to suffer from the limitations asserted by the Applicant, as move instructions are not swap instruction.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-18 are rejected under 35 U.S.C. 102(b) as being unpatentable over Hennessy et al (Computer Architecture: A Quantitative Approach), herein referred to as Hennessy et al.

As per **claim 1**, Hennessy et al discloses a method for processing a plurality of swap requests (See sections 3.3-3.4: Hennessy et al teaches situation where data hazards may come from and ways to handle those hazards) comprising:

receiving a first swap request in a pipeline wherein the first swap request requests swapping active contents of an active register window with a first contents from a first register (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVI2S moves contents from a first register to a temporary register, or active register window);

executing the first swap request (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The DLX pipeline can handle instructions such as moves) including:

executing a first save operation wherein the active contents of the active register window is saved to a corresponding register (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVI2S moves contents from a first register to a temporary register, or active register window); and

executing a first restore operation, wherein the first contents of the first register are restored to the active register window (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVS2I moves contents from a temporary register, or active register window to the first register);

receiving a second swap request in the pipeline immediately subsequent to the first swap request (See section 3.3. figure 3.5, page 142: and section 2.8, figure 2.25, page 104: The second swap is Instruction 2 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping), wherein the second swap request requests swapping the first contents in the active register window with a second contents from a second register (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVS2I moves contents from a temporary register, or active register window to the second register);

determining if the first register is a same register as the second register (See sections 3.3-3.4: Determining if the first swap request and the second swap request swap a same register is part of the hazard detection that must go on in order to properly prevent hazards in the pipeline); and

executing the second swap request if the first swap request and the second swap request do not swap the same register (See section 3.3, figure 3.6, page 142: It would be as if the first swap is Instruction 1 and the second swap is Instruction 2, whereas no conflicts/hazards arise), wherein executing the second swap request includes:

executing a second save operation wherein the first contents of the active register window is saved to the first register substantially simultaneously with the executing the first restore operation (See section 3.4, figure 3.12, page 153: With data forwarding a register can be written to and read at the same time); and

executing a second restore operation, wherein the second contents of the second register are restored to the active register window (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping), wherein the first swap request and the second swap request have a constant latency (See section 3.3, figure 3.6, page 142: It would be as if the first swap is Instruction 1 and the second swap is Instruction 2, whereas no conflicts/hazards arise).

As per **claim 5**, Hennessy et al discloses further comprising: delaying execution of the second swap request if the first register is the same register as the second register (See section 3.3, figure 3.7, page 143: The delaying of an execution is represented by the stall); and executing the second swap request (See section 3.3, figure 3.7, page 143: The instruction starts execution after the stall).

As per **claim 6**, Hennessy et al discloses wherein the execution of the second swap request is delayed sufficiently to allow the execution of the first swap request to be completed (See section 3.3, figure 3.7, page 139 and 143 and page: Stalling is done as long as it is necessary to ensure proper execution of instructions).

As per **claim 7**, Hennessy et al discloses wherein the execution of the second swap request is delayed a predetermined number of clock cycles (See section 3.3, figure 3.6, page 142, and figure 3.7, page 143: Lengths of stalls are based on number of

clock cycles and on a basic 5 stage pipeline, as shown in the DLX model, a stall can take up as much as 3 cycles. The type of stall determines the length of a stall).

As per **claim 8**, Hennessy et al discloses wherein the execution of the second swap request is delayed one clock cycle (See section 3.4, figure 3.13, page 154 and pages 152-155: The figure shows where with data forwarding a stall can be limited to just one clock cycle and doing multiple moves is a case where data hazards require stalls since the pipeline must wait until the data is about to be written to a register before it can be forwarded to the next instruction).

As per **claim 9**, Hennessy et al discloses wherein the pipeline includes more than one processing thread (See section 3.7, pages 187-199: The basic DLX pipeline is extended to include multiple stages of execution (as shown in figure 3.44 on page 190) meaning that instructions are split up among the different execution stages. Those groups of split up instructions are known as threads).

As per **claim 10**, Hennessy et al discloses wherein determining if the first register is the same register as the second register includes determining if the first register in the corresponding processing thread is the same register as the second register in the corresponding processing thread (See section 3.7, pages 187-199: An instruction in a thread will include a register).

As per **claim 11**, Hennessy et al discloses wherein determining if the first register is the same register as the second register occurs as the second swap request is received (See section 3.4, figure 3.13, page 154: The stall bubble does not occur until instruction is received an decoded).

As per **claim 12**, Hennessy et al discloses a method for processing a plurality of consecutive swap requests (See sections 3.3-3.4: Hennessy et al teaches situation where data hazards may come from and ways to handle those hazards) in a multithreaded microprocessor pipeline (See section 3.7, pages 187-199: The basic DLX pipeline is extended to include multiple stages of execution (as shown in figure 3.44 on page 190), meaning that instructions are split up among the different execution stages. Those groups of split up instructions are known as threads) comprising:

receiving a first swap request in a pipeline wherein the first swap request requests swapping active contents of an active register window with a first contents from a first register (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping);

executing the first swap request (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The DLX pipeline can handle instructions such as moves) including:

executing a first save operation wherein the active contents of the active register window is saved to corresponding register (See section 3.3. figure 3.5,

page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVI2S moves contents from a first register to a temporary register, or active register window); and

executing a first restore operation, wherein the first contents of the first register are restored to the active register window (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVS2I moves contents from a temporary register, or active register window to the first register);

receiving a second swap request in the pipeline (See section 3.3. figure 3.5, page 142: and section 2.8, figure 2.25, page 104: The second swap is Instruction 2 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping), wherein the second swap request requests swapping the first contents in the active register window with a second contents from a second register (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVS2I moves contents from a temporary register, or active register window to the second register);

determining if the first register in the corresponding processing thread is a same register as the second register in the corresponding processing thread (See sections 3.3-3.4: Determining if the first swap request and the second swap request swap a

same register is part of the hazard detection that must go on in order to properly prevent hazards in the pipeline); and

executing the second swap request if the first swap request and the second swap request do not swap the same register (See section 3.3, figure 3.6, page 142: It would be as if the first swap is Instruction 1 and the second swap is Instruction 2, whereas no conflicts/hazards arise), wherein executing the second swap request includes:

executing a second save operation wherein the first contents of the active register window is saved to the first register substantially simultaneously with the executing the first restore operation (See section 3.4, figure 3.12, page 153: With data forwarding a register can be written to and read at the same time); and

executing a second restore operation, wherein the second contents of the second register are restored to the active register window (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping), wherein the first swap request and the second swap request have a constant latency (See section 3.3, figure 3.6, page 142: It would be as if the first swap is Instruction 1 and the second swap is Instruction 2, whereas no conflicts/hazards arise).

As per **claim 14**, Hennessy et al discloses further comprising: delaying execution of the second swap request at least one clock cycle if the first register in the corresponding processing thread is a same register as the second register in the corresponding processing thread (See section 3.3, figure 3.7, page 143: The delaying

of an execution is represented by the stall); and executing the second swap request (See section 3.3, figure 3.7, page 143: The instruction starts execution after the stall).

As per **claim 15**, Hennessy et al discloses a plurality of pipeline registers (See section 2.8, page 98: There are 32 registers), at least one of the plurality of pipeline registers being capable of comparing a first swap request and a second swap request (See sections 3.3-3.4: Determining if the first swap request and the second swap request swap a same register is part of the hazard detection that must go on in order to properly prevent hazards in the pipeline and any register is capable of doing a move);

a plurality of active registers (See section 2.8, page 98: The floating point registers can be used as active registers); and

computer readable code stored on a computer readable medium for receiving a first swap request in a pipeline wherein the first swap request requests swapping active contents of an active register window with a first contents from a first register (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVI2S moves contents from a first register to a temporary register, or active register window);

computer readable code stored on a computer readable medium for executing the first swap request (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The DLX pipeline can handle instructions such as moves) including:

executing a first save operation wherein the active contents of the active register window is saved to a corresponding register (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVI2S moves contents from a first register to a temporary register, or active register window); and

executing a first restore operation, wherein the first contents of the first register are restored to the active register window (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVS2I moves contents from a temporary register, or active register window to the first register);

computer readable code stored on a computer readable medium for receiving a second swap request in the pipeline immediately subsequent to the first swap request (See section 3.3. figure 3.5, page 142: and section 2.8, figure 2.25, page 104: The second swap is Instruction 2 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping), wherein the second swap request requests swapping the first contents in the active register window with a second contents from a second register (See section 3.3. figure 3.6, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping. MOVS2I moves contents from a temporary register, or active register window to the second register);

computer readable code stored on a computer readable medium for determining if the first register is a same register as the second register (See sections 3.3-3.4: Determining if the first swap request and the second swap request swap a same register is part of the hazard detection that must go on in order to properly prevent hazards in the pipeline); and

computer readable code stored on a computer readable medium for executing the second swap request if the first swap request and the second swap request do not swap the same register (See section 3.3, figure 3.6, page 142: It would be as if the first swap is Instruction 1 and the second swap is Instruction 2, whereas no conflicts/hazards arise), wherein executing the second swap request includes:

executing a second save operation wherein the first contents of the active register window is saved to first register at substantially simultaneously with the executing the first restore operation (See section 3.4, figure 3.12, page 153: With data forwarding a register can be written to and read at the same time); and

executing a second restore operation, wherein the second contents of the second register are restored to the active register window (See section 3.3. figure 3.5, page 142 and section 2.8, figure 2.25, page 104: The first swap is Instruction 1 and the various move instructions, such as MOVI2S, MOVS2I, etc, are instructions for swapping), wherein the first swap request and the second swap request have a constant latency (See section 3.3, figure 3.6, page 142: It would be as if the first swap is Instruction 1 and the second swap is Instruction 2, whereas no conflicts/hazards arise).

(Examiner's note: The disclosure of the methodology by Hennessy et al inherently discloses the logic for performing the operations as the disclosure is describing computer architecture. It is also further evidenced with the various drawings of the pipelines where such operations would occur and the logic to perform such operations can be found).

As per **claim 16**, Hennessy et al discloses wherein the plurality of pipeline registers includes at least eight pipeline registers, and wherein the at least eight pipeline registers are linked to one of the plurality of active registers (See section 2.8, page 98: There are 32 general purpose registers and a plurality of floating point registers which can be used as pipeline registers and active registers, respectively).

As per **claim 17**, Hennessy et al discloses wherein the plurality of pipeline registers includes 32 pipeline registers (See section 2.8, page 98: There are 32 general purpose registers).

As per **claim 18**, Hennessy et al discloses wherein the pipeline architecture is one of at least two pipeline architectures in a single multithreaded microprocessor (See section 3.7, figure 3.44, page 190: The basic DLX pipeline is extended to include multiple stages of execution).

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Vincent Lai whose telephone number is (571) 272-6749. The examiner can normally be reached on M-F 8:00-5:30 (First BiWeek Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on (571) 272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Vincent Lai
Examiner
Art Unit 2181

vl
March 30, 2007

DONALD SPARKS
SUPERVISORY PATENT EXAMINER